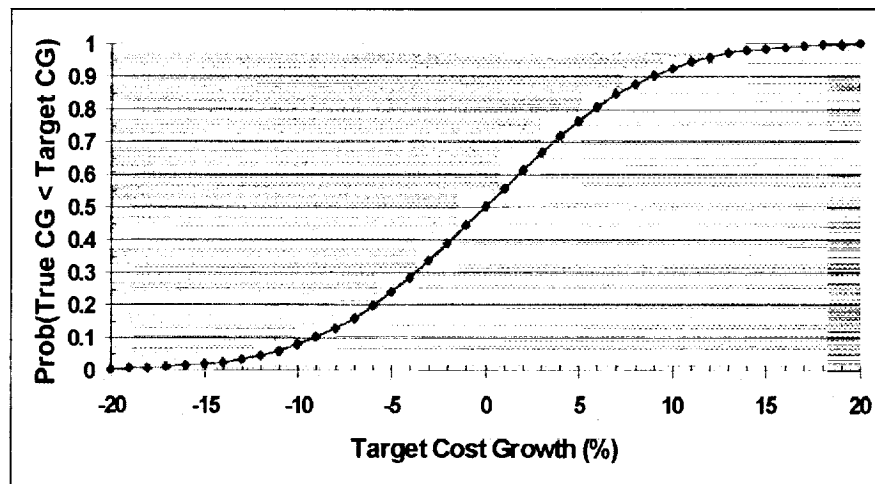


Systems Engineering Programmatic Estimation Using Technology Variance

(H-32236D)



$$P(CG < TCG) = f(TCG, \mu_{NN}, \sigma_{NN}, \mu_C, \sigma_C)$$

Final Report to NASA-MSFC

By

OR Applications

February 7, 2000

Systems Engineering Programmatic Estimation Using Technology Variance

TABLE OF CONTENTS

1.0 INTRODUCTION	1
1.1 STUDY OBJECTIVE	1
1.2 EXPECTED SIGNIFICANCE	2
1.3 STATEMENT OF UNIQUENESS	2
2.0 TECHNICAL RESULTS	2
2.1 DERIVATION/IMPLEMENTATION OF TECHNOLOGY RETURN AND VARIANCE PARAMETERS	2
2.1.1 <i>Current Prediction Method</i>	2
2.1.2 <i>Expression of Probability Statement Using Technology Return and Variance</i>	3
2.1.3 <i>Application to MSRR-1 Pilot Study</i>	5
3.0 CONCLUSIONS	6
4.0 REFERENCES	7
APPENDIX: NEURAL NET AND REGRESSION MODELS	9

Systems Engineering Programmatic Estimation Using Technology Variance

Abstract

Unique and innovative system programmatic estimation is conducted using the variance of the packaged technologies. Covariance analysis is performed on the subsystems and components comprising the system of interest. Technological "return" and "variation" parameters are estimated. These parameters are combined with the model error to arrive at a measure of system development stability. The resulting estimates provide valuable information concerning the potential cost growth of the system under development.

1.0 Introduction

Systems engineering (SE) is a field that is generally viewed as the process of formulating and solving problems at a very high conceptual level. SE covers all phases of the life cycle of a system. One phase of particular interest is that of system development. The conceptual design of a system may or may not resemble the design at the end of the system development phase. Moreover, the cost and schedule parameters established at the beginning of development can vary significantly from those at system delivery. However, the basic technologies for the final design are usually established fairly early on in the SE process. What information can be learned from these technologies early on that will be useful for long-term prediction in the system development phase?

The Complex Organizational Metric for Programmatic Risk Environments (COMPRÉ) tool utilizes technology return and variance in synthesizing programmatic elements into predictions of **expected** cost growth as a measure of programmatic success. However, COMPRÉ does not directly utilize technology return and variance in arriving at **probability** statements concerning programmatic cost growth.

1.1 Study Objective

The objective of this study is to improve the understanding of the role of technology return and variance parameters in predicting probability of success for system development programs. This effort will help answer the following key questions:

1. Can technology return and variance parameters be used to gain insight into systems development measures of performance, specifically probability of cost and schedule growth?
2. If so, what parameters are significant, and how can they be used?
3. Do the results satisfy face validity?

1.2 Expected Significance

The expected significance of this research and development effort includes an improved understanding of systems engineering design and development through the quantification of probability measures of performance relating to technology return and variance parameters.

1.3 Statement of Uniqueness

The combination of technology return and variance parameters with neural networks to produce meaningful probability statements has not been previously applied to large complex programs of the type that NASA frequently encounters.

2.0 Technical Results

2.1 Derivation/Implementation of Technology Return and Variance Parameters

In this section, the method for utilizing technology return and variance parameters in the prediction of probability of success for programs is developed. The method of application is implemented and verified using the available database and existing Pilot Study.

2.1.1 Current Prediction Method

Figure 2.1.1-1 provides the current configuration of the COMPRE model. Results from the covariance analysis feed directly into the artificial neural network. This neural net predicts expected cost growth. The expected cost growth, in combination with the neural net model error, results in an "s-curve" prediction of probability of cost growth. Note, in particular, that there is no direct link between the covariance analysis and the "s-curve."

The mathematics of the "s-curve" prediction show clearly this lack of a direct link. The equation for prediction is given by:

$$P(CG < TCG) = NORMS\left(\frac{TCG - \mu_{NN}}{\sigma_{NN}}\right)$$

where,

CG = actual (unknown) programmatic cost growth (%),

NORMS = standard normal distribution,

P = probability,

TCG = target cost growth (%),

μ_{NN} = expected cost growth (%) from neural net,

σ_{NN} = neural net average error (%) in predicting cost growth.

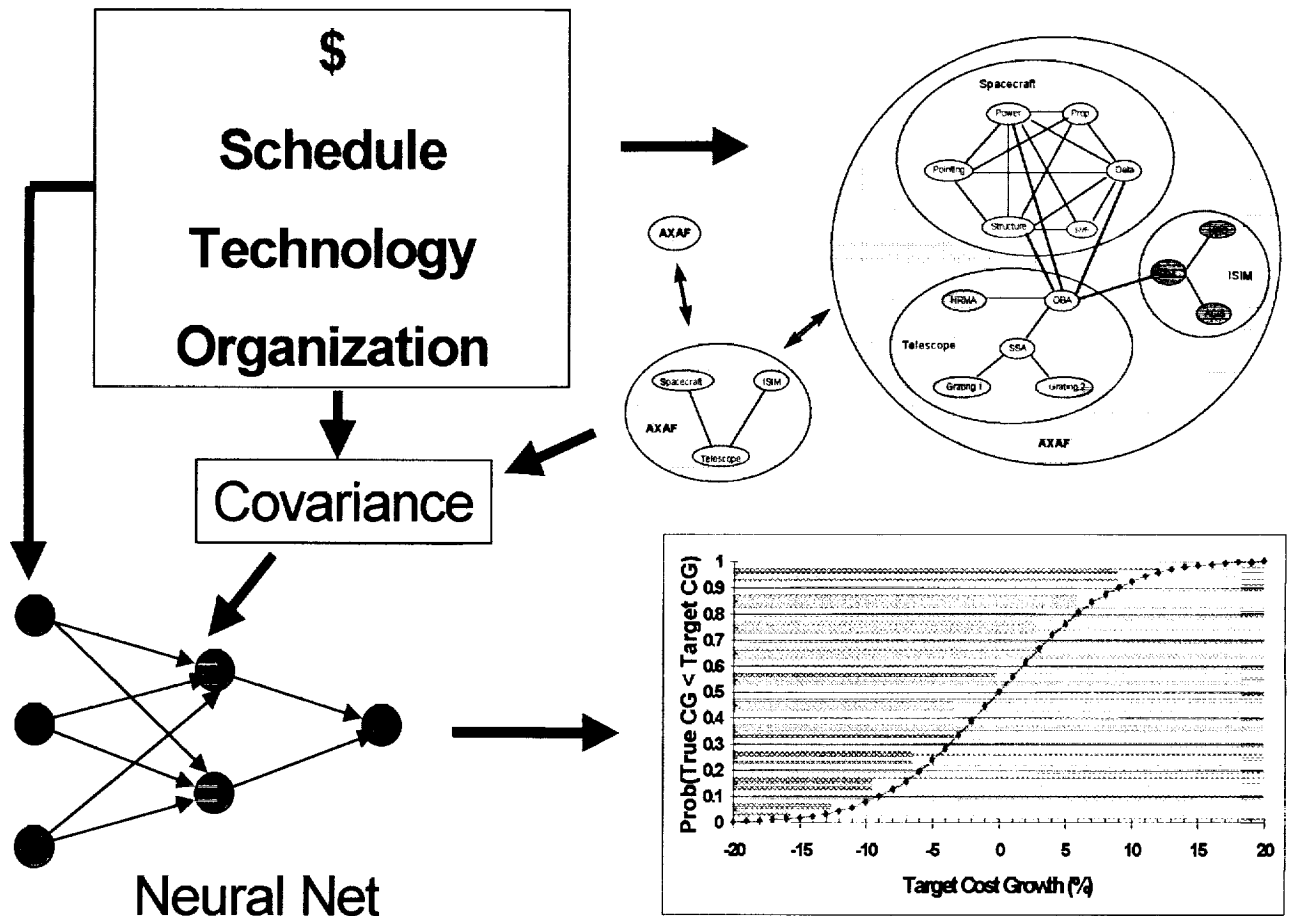


Figure 2.1.1-1. Current COMPRE Methodology Configuration

2.1.2 Expression of Probability Statement Using Technology Return and Variance

A direct link must be provided between the covariance analysis of Figure 2.1.1-1 and the "s-curve" probability statement of the same figure. Thus, the probability of cost growth would depend not only on the output and modeling error of the neural net, but also on the "return" and variance (more specifically, standard deviation) of the technology package utilized in the construction of the system. The general equation for such a linked relationship is given below. It was the purpose of this study to determine and evaluate the form of the function, f , in this relationship. An important consideration in determining the form of f is in the assumption of dependency among the "means" and "standard deviations"

involved in this mathematical expression. For instance, are the neural net and covariance standard deviations independent of one another? What about the means?

$$P(CG < TCG) = f(TCG, \mu_{NN}, \sigma_{NN}, \mu_C, \sigma_C)$$

where,

CG = actual (unknown) programmatic cost growth (%),

f = function,

P = probability,

TCG = target cost growth (%),

μ_C = expected value from covariance analysis,

μ_{NN} = expected cost growth (%) from neural net,

σ_C = standard deviation from covariance analysis,

σ_{NN} = neural net average error (%) in predicting cost growth.

In fact, we may write the desired probability statement as the standard normal distribution calculation:

$$P(CG < TCG) = NORMS\left(\frac{TCG - \mu_T}{\sigma_T}\right)$$

where,

CG = actual (unknown) programmatic cost growth (%),

NORMS = standard normal distribution,

P = probability,

TCG = target cost growth (%),

μ_T = expected total cost growth (%),

σ_T = total average error (%) in predicting cost growth.

Then, the question becomes how to calculate the total mean and standard deviation as a function of those from the covariance analysis and neural net. If we merely were in a situation where the sum of the random variables (from the covariance and neural net) were of interest, then we would write:

$$\mu_T = \mu_C + \mu_{NN}$$

$$\sigma_T^2 = \sigma_C^2 + \sigma_{NN}^2 + 2\rho_{C,NN}\sigma_C\sigma_{NN}$$

where, again, the subscripts C and NN refer to covariance and neural net, respectively, and $\rho_{C,NN}$ is the correlation between the covariance and neural net data. However, we are not in a situation in which addition of the random variables is appropriate, since the

covariance analysis is an *intermediate* step in calculating the neural net results. Thus, we are in a situation in which measurement error is the key consideration. The prediction of cost growth from the neural net is still the best available estimate of expected value, but the neural net error is only one contributor to the standard deviation of that estimated cost growth. Thus, we see from Juran that:

$$\begin{aligned}\mu_T &= \mu_{NN} \\ \sigma_T^2 &= \sigma_C^2 + \sigma_{NN}^2 + \rho_{C,NN} \sigma_C \sigma_{NN}\end{aligned}$$

Now, we must determine how to calculate the correlation, $\rho_{C,NN}$. We may estimate this correlation as the correlation between the covariance expected value and the neural net expected value taken over the entire database of programs:

$$\begin{aligned}\rho_{C,NN} &= \rho(\mu_C, \mu_{NN}) \\ \text{or} \\ \rho_{C,NN} &= 0.628\end{aligned}$$

Then, our final form becomes:

$$\begin{aligned}\mu_T &= \mu_{NN} \\ \sigma_T^2 &= \sigma_C^2 + \sigma_{NN}^2 + 0.628 \sigma_C \sigma_{NN} \\ P(CG < TCG) &= NORMS\left(\frac{TCG - \mu_{NN}}{\sqrt{\sigma_C^2 + \sigma_{NN}^2 + 0.628 \sigma_C \sigma_{NN}}}\right)\end{aligned}$$

2.1.3 Application to MSRR-1 Pilot Study

For the MSRR-1 program, we have the following pertinent values:

$$\begin{aligned}\mu_{NN} &= 12.6 \%, \\ \sigma_C &= 13.64 \%, \\ \sigma_{NN} &= 5.0 \%\end{aligned}$$

Substituting into the equation above, we get:

$$P(CG < TCG) = NORMS\left(\frac{TCG - 12.6}{15.93}\right)$$

From Figure 2.1.3-1, we see that the 95% confidence level for cost growth is approximately 39%. The 50% confidence level is, of course, equal to the expected value of the neural net output, or a cost growth of 12.6%.

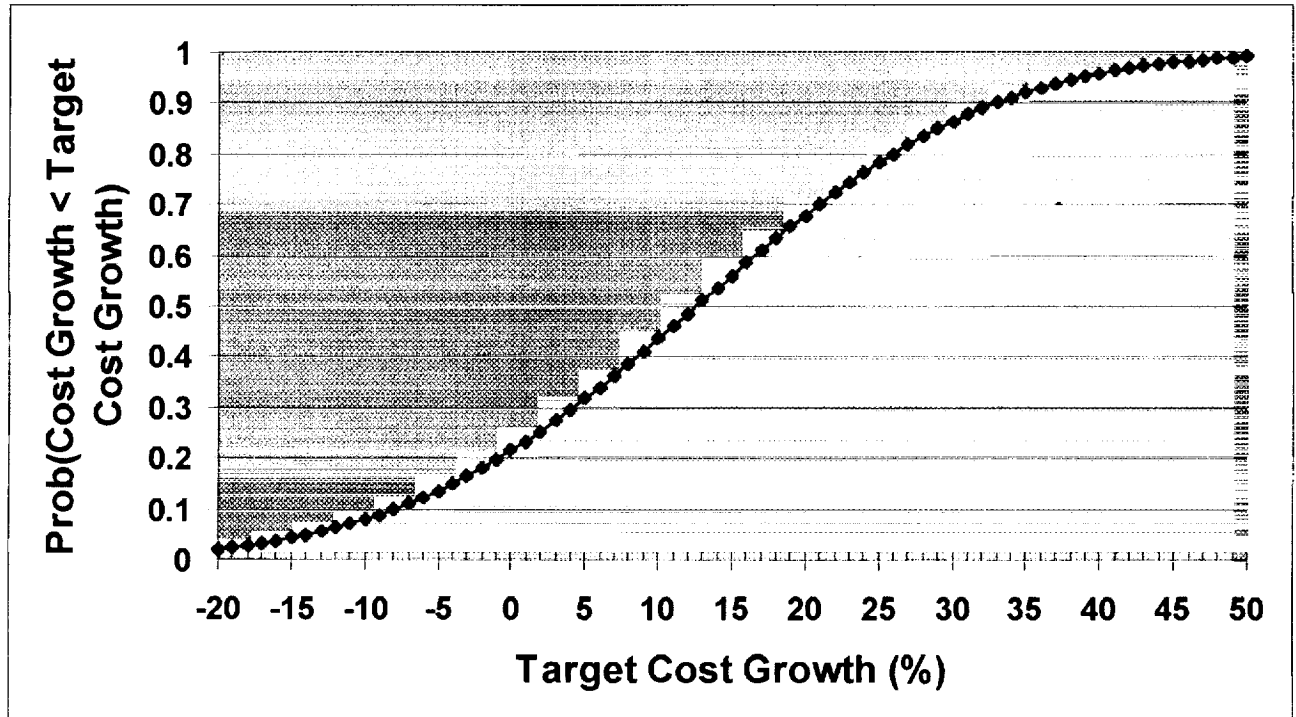


Figure 2.1.3-1. MSRR-1 Cost Growth Profile

3.0 Conclusions

1. The neural net estimate for expected cost growth of the program of interest remains the best estimate for same. This estimate is unaltered by the presence of the covariance analysis estimate for cost growth, since that estimate (and its correlation with the neural net output) are accounted for directly in the final neural net estimate for cost growth.
2. The standard deviation for programmatic cost growth is arrived at by including both the neural net error and the covariance estimate for standard deviation. This is so, because the neural net does not predict standard deviation directly, and, therefore, our probability calculations must consider both sources of variation as well as their correlation.

3. The MSRR-1 expected value and 95% confidence estimates for cost growth are significantly altered by this revision to the COMPRE methodology.

4.0 References

- Bunde, A., and Havlin, S., Fractals in Science, Springer-Verlag, 1994.
- Cameron, K.S., and Whetten, D.A., Organizational Effectiveness: A Comparison of Multiple Models, Academic Press, 1983.
- Chartrand, G., Introductory Graph Theory, Dover Publications, Inc., 1977.
- Crouch, E.A.C., and Wilson, R., Risk/Benefit Analysis, Ballinger Publishing Co., 1982.
- Draper, N., and Smith, H., Applied Regression Analysis, John Wiley & Sons, 1981.
- Drouin, M., Abou-Kandil, H., and Mariton, M., Control of Complex Systems: Methodology and Technology, Plenum Press, 1991.
- Efron, B., "Bootstrap Methods: Another Look at the Jackknife," *The Annals of Statistics*, Vol. 7, No. 1, 1979.
- Freudenberg, W.R., and Rursch, J.A., "The Risks of 'Putting the Numbers in Context': A Cautionary Tale," *Risk Analysis*, Vol. 14, No. 6, 1994.
- Flood, R.L., and Carson, E.R., Dealing With Complexity: An Introduction to the Theory and Application of Systems Science, Plenum Press, 1988.
- Hastings, H.M, and Sugihara, G., Fractals: A User's Guide for the Natural Sciences, Oxford University Press, 1995.
- Harvard School of Public Health, "Analyzing Risk: Science, Assessment, and Management," September, 1994.
- Hertz, J., Krogh, A., and Palmer, R.G., Introduction to the Theory of Neural Computation, Addison-Wesley, 1991.
- Hudak, D.G., "Adjusting Triangular Distributions for Judgmental Bias," *Risk Analysis*, Vol. 14, No. 6, 1994.
- Juran, J.M., Juran's Quality Control Handbook, McGraw-Hill, Inc., 1988.
- Klir, G.J., Architecture of Systems Problem Solving, Plenum Press, 1985.

Kronsjö, L., Computational Complexity of Sequential and Parallel Algorithms, John Wiley & Sons, 1985.

Lee, S.M.S., and Young, G.A., "Sequential Iterated Bootstrap Confidence Intervals," *Journal of the Royal Statistical Society*, Vol. 58, No. 1, 1996.

Lloyd, S., "Complexity Simplified," *Scientific American*, May 1996.

Mog, R.A., "Systems Engineering Metrics: Organizational Complexity and Product Quality Modeling," H-27289D, Final Report, OR Applications, September 1997.

Mog, R.A., "Risk Assessment Methodologies for Spacecraft in Particulate Environments: Enhancements Through Operations Research," OR Applications Final Report, September 1996.

Mog, R.A., "The Use of the Poisson Process in Environmental Risk Assessment," Internal Memo, July 1994.

Nemhauser, G.L., and Kan, A.H.G., Handbooks in Operations Research and Management Science: Computing, Vol. 3, North-Holland, 1992.

Nemhauser, G.L., Kan, A.H.G., and Todd, M.J., Handbooks in Operations Research and Management Science: Optimization, Vol. 1, North-Holland, 1989.

Neuts, M.F., Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach, Dover Publications, Inc., 1981.

Nicolis, J.S., Chaos and Information Processing: A Heuristic Outline, World Scientific, 1991.

Papadimitriou, C.H., Computational Complexity, Addison-Wesley, 1994.

Peliti, L., and Vulpiani, A., Measures of Complexity, Springer-Verlag, 1988.

Sedgewick, R., Algorithms, Addison-Wesley, 1988.

Society for Risk Analysis, *Risk Analysis: Uncertainty Analysis*, Vol. 14, No. 4, August 1994.

Streufert, S., and Swezey, R.W., Complexity, Managers, and Organizations, Academic Press, Inc., 1986.

Temkin, O.N., Zeigarnik, A.V., and Bonchev, D., Chemical Reaction Networks: A Graph-Theoretical Approach, CRC Press, Inc., 1996.

Thomas, D., and Mog, R., "A Quantitative Metric of System Development Complexity," 7th Annual INCOSE 1997 Symposium, August, 1997.

Thomas, D., Private Communication, August 12, 1996.

Tsoukalas, L. H., and Uhrig, R. E., Fuzzy and Neural Approaches in Engineering, John Wiley & Sons, Inc., 1997.

APPENDIX: Neural Net and Regression Models

A large number of neural net and regression models have been developed in this effort. The regression models take the form:

$$CG = \sum_{i=1}^n k_i C^{\alpha_i} t^{\beta_i} \sigma^{\delta_i} \lambda^{\gamma_i} \chi^{\mu_i}$$

where,

C = program cost (1996 \$M),

CG = program cost growth (%),

N = number of terms in regression equation,

t = program duration (years),

χ = program chromatic number,

λ = COMPRE technology return parameter,

σ = COMPRE technology risk parameter,

and all other symbols are estimated parameters.

Note that for $n = 1$, the resulting form is intrinsically linear through logarithmic transformation. All other values for n result in an intrinsically nonlinear regression form. One intrinsically linear form, which was found to be statistically significant is:

$$CG = t^{1.673} \sigma^{2.767} \lambda^{-3.242} - 16.8$$

This regression model is intuitively plausible, in that one would expect cost growth to tend to increase with program duration and technology risk, and possibly decrease with technology return. While the significance level for this model is 0.01, the correlation coefficient is $R = 0.57$, showing that only about a third of the total statistical variation in the data has been explained by this model. Furthermore, the average error (in cost growth) for this model is 34.2% per program, which is unacceptably high for this author. Figure A-1 shows the fit of predicted to actual cost growth for the 31 programs in the database.

A large number of intrinsically nonlinear models ($n > 1$) were also developed, but none achieved average errors less than 23% per program.

A number of neural net models were trained to the database of interest. Table A-1 shows the connection weights for the best model. This model has a correlation of 0.97, and an average error of 5% per program. It also fits more lower cost growth (less than 20%) programs better than previously considered neural net models. Finally, this model is superior to previous neural net models which involved transformations (sometimes called squashing) to the covariance outputs prior to neural net input.

Although several programs are not well-fit (see Figure A-2), this is consistent with the presumption that some programs have higher (or lower) achieved cost growth than their respective cost, technology, and schedule parameters would indicate, even given perfect information.

Finally, unlike previously considered neural nets, this neural net is also fully consistent with the technology risk and return parameters resulting from the COMPRE covariance analysis process. The neural net approach outperforms even intrinsically nonlinear regression models (which also require a high degree of parameter training), because neural nets synthesize information like a simultaneous system of equations, whereas the regression approach involves estimating parameters for a single equation, which may not be sufficient to capture complex phenomena.

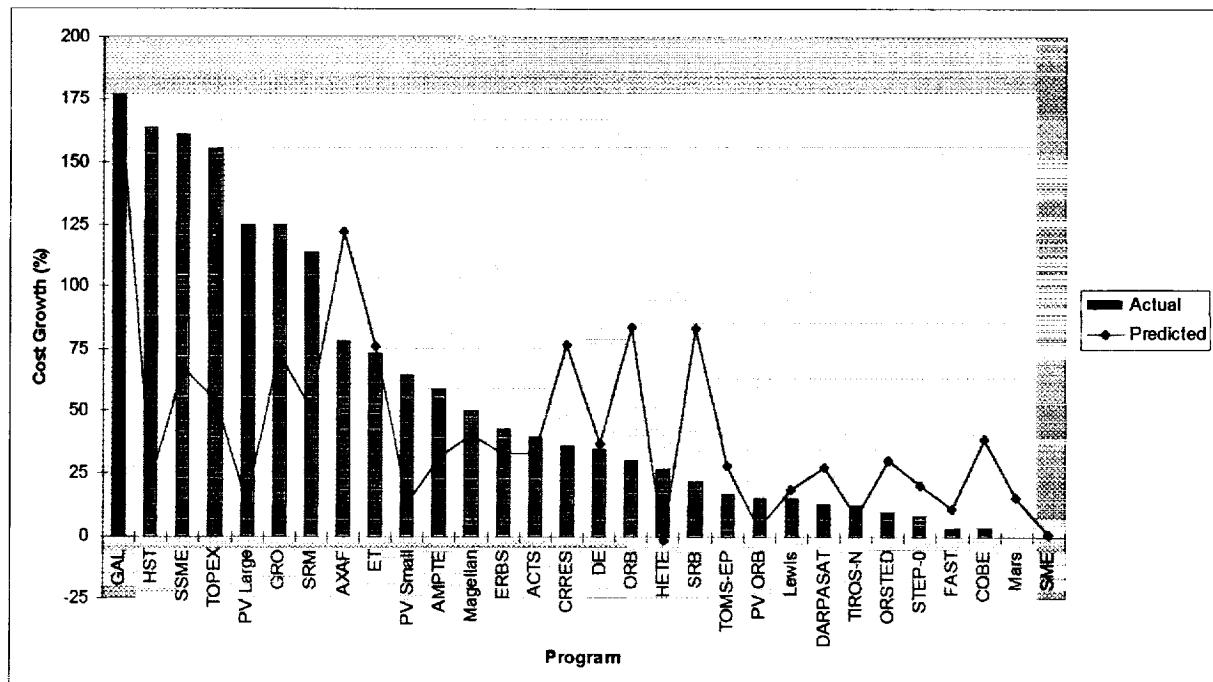


Figure A-1. Cost Growth Fit for Intrinsically Linear Model

Table A-1. COMPRE Neural Net Connection Weights

Hidden Layer or Output Node	Cost Variable	Duration Variable	COMPRE Risk Variable (sigma)	COMPRE Return Variable (lambda)	Chromatic Number Variable (chi)	Bias Node	Cost Growth (output node)
1	40.35	2.20	40.82	43.96	1.71	-12.40	17.40
2	-5.29	-33.40	19.26	29.63	-26.92	6.74	6.58
3	4.61	-4.20	4.83	-9.30	6.61	-12.61	0.64
4	0.72	18.33	4.67	-19.73	-9.09	-2.57	10.64
5	-8.72	17.75	-0.85	5.67	-46.66	2.90	-5.96
Cost Growth (output)	N/A	N/A	N/A	N/A	N/A	-19.99	N/A

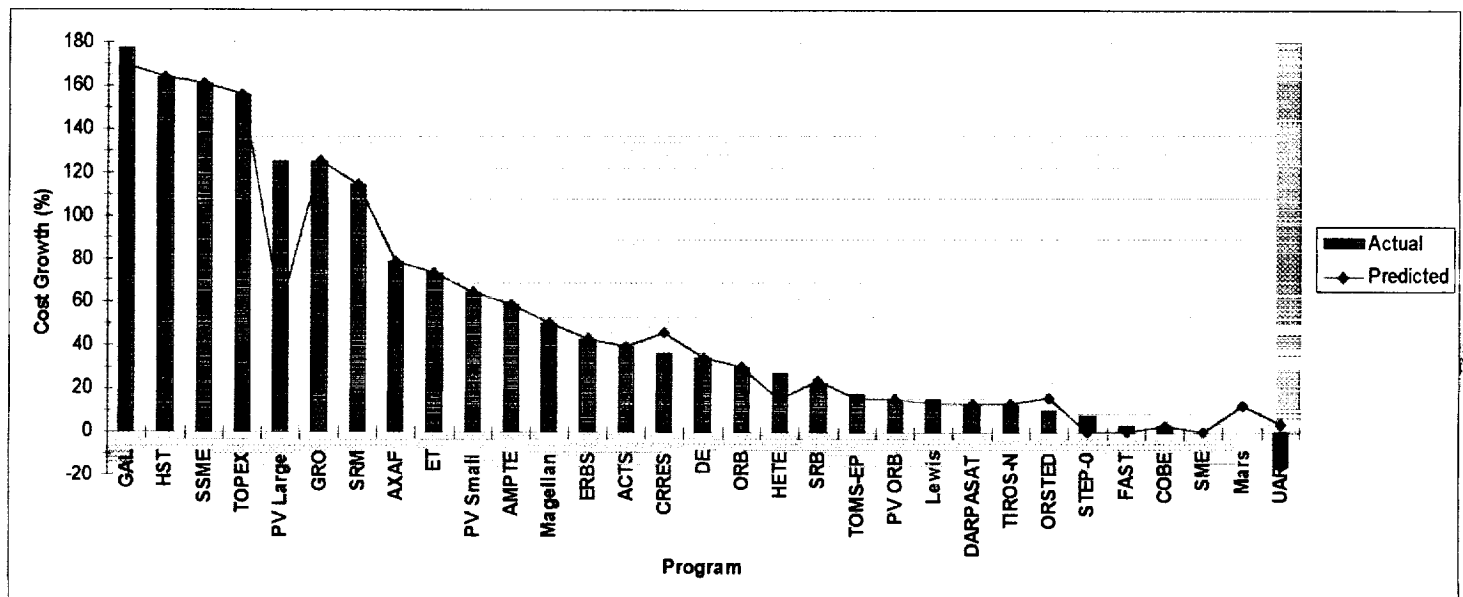


Figure A-2. Cost Growth Fit for Neural Net

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE February 2000		3. REPORT TYPE AND DATES COVERED Final Report
4. TITLE AND SUBTITLE Systems Engineering Programmatic Estimation Using Technology Variance			5. FUNDING NUMBERS C - H32236D	
6. AUTHORS Robert A. Mog, Ph.D.				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) OR Applications 7811 Lent Drive Huntsville, AL 35802			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) NASA-MSFC George C. Marshall Space Flight Center MSFC, AL 35812			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unlimited Distribution			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) <i>Unique and innovative system programmatic estimation is conducted using the variance of the packaged technologies. Covariance analysis is performed on the subsystems and components comprising the system of interest. Technological "return" and "variation" parameters are estimated. These parameters are combined with the model error to arrive at a measure of system development stability. The resulting estimates provide valuable information concerning the potential cost growth of the system under development.</i>				
14. SUBJECT TERMS Systems Engineering, Metrics, Covariance, Programmatic, Risk Assessment, Artificial Neural Networks, Regression.			15. NUMBER OF PAGES 12	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	